# Automatic Acquisition of Huge Training Data for Bio-Medical Named Entity Recognition

**Yu Usami**[*] [†] **Han-Cheol Cho**[†] **Naoaki Okazaki**[‡] **and Jun'ichi Tsujii**[§]
[*] Aizawa Laboratory, Department of Computer Science, The University of Tokyo, Tokyo, Japan
[†] Tsujii Laboratory, Department of Computer Science, The University of Tokyo, Tokyo, Japan
[‡] Inui Laboratory, Department of System Information Sciences, Tohoku University, Sendai, Japan
[§] Microsoft Research Asia, Beijing, China
{yusmi, hccho}@is.s.u-tokyo.ac.jp
okazaki@ecei.tohoku.ac.jp
jtsujii@microsoft.com

## Abstract

Named Entity Recognition (NER) is an important first step for BioNLP tasks, e.g., gene normalization and event extraction. Employing supervised machine learning techniques for achieving high performance recent NER systems require a manually annotated corpus in which every mention of the desired semantic types in a text is annotated. However, great amounts of human effort is necessary to build and maintain an annotated corpus. This study explores a method to build a high-performance NER without a manually annotated corpus, but using a comprehensible lexical database that stores numerous expressions of semantic types and with huge amount of unannotated texts. We underscore the effectiveness of our approach by comparing the performance of NERs trained on an automatically acquired training data and on a manually annotated corpus.

## 1 Introduction

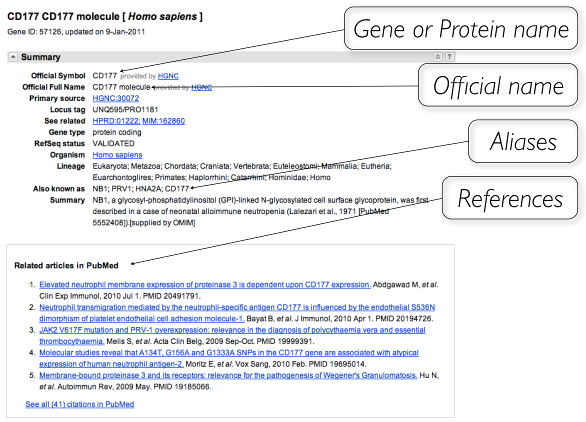
Named Entity Recognition (NER) is the task widely used to detect various semantic classes such as genes (Yeh et al., 2005), proteins (Tanabe and Wilbur, 2002), and diseases in the biomedical field.

A naíve approach to NER handles the task as a dictionary-matching problem: Prepare a dictionary (gazetteer) containing textual expressions of named entities of specific semantic types. Scan an input text, and recognize a text span as a named entity if the dictionary includes the expression of the span.

Although this approach seemingly works well, it presents some critical issues. First, the dictionary must be comprehensive so that every NE mention can be found in the dictionary. This requirement for dictionaries is stringent because new terminology is being produced continuously, especially in the biomedical field. Second, this approach might suffer from an ambiguity problem in which a dictionary includes an expression as entries for multiple semantic types. For this reason, we must use the context information of an expression to make sure that the expression stands for the target semantic type.

Nadeau and Sekine (2007) reported that a strong trend exists recently in applying machine learning (ML) techniques such as Support Vector Machine (SVM) (Kazama et al., 2002; Isozaki and Kazawa, 2002) and Conditional Random Field (CRF) (Settles, 2004) to NER, which can address these issues. In this approach, NER is formalized as a classification problem in which a given expression is classified into a semantic class or other (non-NE) expressions. Because the classification problem is usually modeled using supervised learning methods, we need a manually annotated corpus for training NER classifier. However, preparing manually annotated corpus for a target domain of text and semantic types is cost-intensive and time-consuming because human experts are needed to reliably annotate NEs in text. For this reason, manually annotated corpora for NER are often limited to a specific domain and covers a small amount of text.

In this paper we propose a novel method for automatically acquiring training data for NER from a comprehensible lexical database and huge amounts of unlabeled text. This paper presents four contribu-



**CD177 CD177 molecule [ *Homo sapiens* ]**

Gene ID: 57126, updated on 9-Jan-2011

**Summary**

| | |
|---|---|
| **Official Symbol** | CD177 provided by HGNC |
| **Official Full Name** | CD177 molecule provided by HGNC |
| **Primary source** | HGNC:30072 |
| **Locus tag** | UNQ595/PRO1181 |
| **See related** | HPRD:01222; MIM:162860 |
| **Gene type** | protein coding |
| **RefSeq status** | VALIDATED |
| **Organism** | Homo sapiens |
| **Lineage** | Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae; Homo |
| **Also known as** | NB1; PRV1; HNA2A; CD177 |
| **Summary** | NB1, a glycosyl-phosphatidylinositol (GPI)-linked N-glycosylated cell surface glycoprotein, was first described in a case of neonatal alloimmune neutropenia (Lalezari et al., 1971 [PubMed 5552408]).[supplied by OMIM] |

**Related articles in PubMed**

1. Elevated neutrophil membrane expression of proteinase 3 is dependent upon CD177 expression. Abdgawad M, *et al*. Clin Exp Immunol, 2010 Jul 1. PMID 20491791.
2. Neutrophil transmigration mediated by the neutrophil-specific antigen CD177 is influenced by the endothelial S536N dimorphism of platelet endothelial cell adhesion molecule-1. Bayat B, *et al*. J Immunol, 2010 Apr 1. PMID 20194726.
3. JAK2 V617F mutation and PRV-1 overexpression: relevance in the diagnosis of polycythaemia vera and essential thrombocythaemia. Melis S, *et al*. Acta Clin Belg, 2009 Sep-Oct. PMID 19999391.
4. Molecular studies reveal that A134T, G156A and G1333A SNPs in the CD177 gene are associated with atypical expression of human neutrophil antigen-2. Moritz E, *et al*. Vox Sang, 2010 Feb. PMID 19695014.
5. Membrane-bound proteinase 3 and its receptors: relevance for the pathogenesis of Wegener's Granulomatosis. Hu N, *et al*. Autoimmun Rev, 2009 May. PMID 19185066.

See all (41) citations in PubMed

Figure 1: Example of an Entrez Gene record.

tions:

1. We show the ineffectiveness of a naíve dictionary-matching for acquiring a training data automatically and the significance of the quality of training data for supervised NERs

2. We explore the use of reference information that bridges the lexical database and unlabeled text for acquiring high-precision and low-recall training data

3. We develop two strategies for expanding NE annotations, which improves the recall of the training data

4. The proposed method acquires a large amount of high-quality training data rapidly, decreasing the necessity of human efforts

## 2 Proposed method

The proposed method requires two resources to acquire training data automatically: a comprehensive lexical database and unlabeled texts for a target domain. We chose Entrez Gene (National Library of Medicine, 2005) as the lexical database because it provides rich information for lexical entries and because genes and proteins constitute an important semantic classes for Bio NLP. Entrez Gene consists of more than six million gene or protein records, each of which has various information such as the official gene (protein) name, synonyms, organism, description, and human created references. Figure 1 presents an example of an Entrez Gene record. We created a dictionary by collecting official gene (protein) names and their synonyms from the Entrez Gene records. For unlabeled text, we use the all 2009 release MEDLINE (National Library of Medicine, 2009) data. MEDLINE consists of about ten million abstracts covering various fields of biomedicine and health. In our study, we focused on recognizing gene and protein names within biomedical text.

Our process to construct a NER classifier is as follows: We apply the GENIA tagger (Tsuruoka et al., 2005) to split the training data into tokens and to attach part of speech (POS) tags and chunk tags. In this work, tokenization is performed by an external program that separates tokens by a space, hyphen, comma, period, semicolon, or colon character. Part of speech tags present grammatical roles of tokens, e.g. verbs, nouns, and prepositions. Chunk tags compose tokens into syntactically correlated segments, e.g. verb phrases, noun phrases, and prepositional phrases. We use the IOBES notation (Ratinov and Roth, 2009) to represent NE mentions with label sequences, thereby NER is formalized as a multi-class classification problem in which a given token is classified into IOBES labels. To classify labels of tokens, we use a linear kernel SVM which applies the one-vs.-the-rest method (Weston and Watkins, 1999) to extend binary classification to multi-class classification. Given the $t$-th token $x_t$ in a sentence, we predict the label $y_t$,

$$y_t = \operatorname*{argmax}_y s(y|x_t, y_{t-1}).$$

In this equation, $s(y|x_t, y_{t-1})$ presents the score (sum of feature weights) when the token $x_t$ is labeled $y$. We use $y_{t-1}$ (the label of the previous token) to predict $y_t$, expecting that this feature behaves as a label bigram feature (also called translation feature) in CRF. If the sentence consists of $x_1$ to $x_T$, we repeat prediction of labels sequentially from the beginning ($y_1$) to the end ($y_T$) of a sentence. We used LIBLINEAR (Fan et al., 2008) as an SVM implementation.

Table 1 lists the features used in the classifier modeled by SVM. For each token (“Human” in the example of Table 1), we created several features including: token itself (w), lowercase token (wl), part of speech (pos), chunk tag (chk), character pattern of

| Name | Description | Example Value |
|---|---|---|
| w | token | Human |
| wl | token in small letters | human |
| pos | part of speech | NNP |
| chk | chunk tag | B-NP |
| shape | entity pattern | ULLLL |
| shaped | entity pattern 2 | UL |
| type | token type | InitCap |
| $p_n(n = 1...4)$ | prefix n characters | (H,Hu,Hum,Huma) |
| $s_n(n = 1...4)$ | suffix n characters | (n,an,man,uman) |

Table 1: Example of features used in machine learning process.

token (shape), character pattern designated (shaped), token type (type), prefixes of length $n$ ($p_n$), and suffixes of length $n$ ($s_n$). More precisely, the character pattern of token (shape) replaces each character in the token with either an uppercase letter (U), a lowercase letter (L), or a digit (D). The character pattern designated (shaped) is similar to a shape feature, but the consecutive character types are reduced to one symbol, for example, "ULLLL" (shape) is represented with "UL" (shaped) in the example of Table 1). The token type (type) represents whether the token satisfies some conditions such as "begins with a capital letter", "written in all capitals", "written only with digits", or "contains symbols". We created unigram features and bigram features (excluding wl, $p_n$, $s_n$) from the prior 2 to the subsequent 2 tokens of the current position.

### 2.1 Preliminary Experiment

As a preliminary experiment, we acquired training data using a naíve dictionary-matching approach. We obtained the training data from all 2009 MEDLINE abstracts with an all gene and protein dictionary in Entrez Gene. The training data consisted of nine hundred million tokens. We constructed a NER classifier using only four million tokens of the training data because of memory limitations. For evaluation, we used the Epigenetics and Post-translational Modification (EPI) corpus BioNLP 2011 Shared Task (SIGBioMed, 2011). Only development data and training data are released as the EPI corpus at present, we used both of the data sets for evaluation in this experiment. Named entities in the corpus are annotated exhaustively and belong to a single semantic class, Gene or Gene Product (GGP) (Ohta et al., 2009). We evaluated the performance of the NER on four measures: Accuracy (a), Precision (P), Recall (R), and F1-measure (F1). We used the strict matching criterion that a predicted named entity is correct if and only if the left and the right boundaries are both correct.

| Method | A | P | R | F1 |
|---|---|---|---|---|
| dictionary matching | 92.09 | 39.03 | 42.69 | 40.78 |
| trained on acquired data | 85.76 | 10.18 | 23.83 | 14.27 |

Table 2: Results of the preliminary experiment.

**(a)** It is clear that in culture media of ***AM***, *cystatin C* and *cathepsin B* are present as proteinase–antiproteinase complexes.

**(b)** Temperature in the puerperium is higher in ***AM***, and lower in PM.

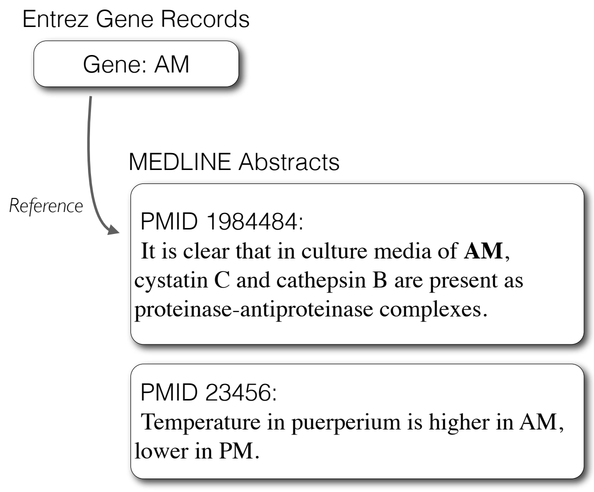
Figure 2: Dictionary-based gene name annotating example (annotated words are shown in italic typeface).

Table 2 presents the evaluation results of this experiment. The first model "dictionary matching" performs exact dictionary-matching on the test corpus. It achieves a 40.78 F1-score. The second model "trained on acquired data" uses the training data acquired automatically for constructing NER classifier. It scores very low-performance (14.27 F1-score), even compared with the simple dictionary-matching NER. Exploring the annotated training data, we investigate why this machine learning approach shows extremely low performance.

Figure 2 presents an example of the acquired training data. The word "AM" in the example (a) is correct because it is gene name, although "AM" in the example (b) is incorrect because "AM" in (b) is the abbreviation of *ante meridiem*, which means before noon. This is a very common problem, especially with abbreviations and acronyms. If we use this noisy training data for learning, then the result of NER might be low because of such ambiguity. It is very difficult to resolve errors in the training data even with the help of machine learning methods.

### 2.2 Using Reference Information

To obtain high-precision data, we used reference information included with each record in Entrez Gene. Figure 3 portrays a simple example of reference information. It shows the reference information of the



PMID 1984484:
It is clear that in culture media of **AM**, cystatin C and cathepsin B are present as proteinase-antiproteinase complexes.

PMID 23456:
Temperature in puerperium is higher in AM, lower in PM.

Figure 3: Reference to MEDLINE abstract example.

Entrez Gene record which describes that the gene “AM”. The reference information indicates PMIDs in which the gene or protein is described.

We applied the rule whereby we annotated a dictionary-matching in each MEDLINE abstract only if they were referred by the Entrez Gene records. Figure 3 shows that the gene “AM” has reference to the MEDLINE abstract #1984484 only. Using this reference information between the Entrez Gene record “AM” and the MEDLINE abstract #1984484, we can annotate the expansion “AM” in MEDLINE abstract #1984484 only. In this way, we can avoid incorrect annotation such as example b in Figure 2.

We acquired training data automatically using reference information, as follows:

1. Construct a gene and protein dictionary including official names, synonyms and reference information in Entrez Gene
2. Apply a dictionary-matching on the all MEDLINE abstracts with the dictionary
3. Annotate the MEDLINE abstract only if it was referred by the Entrez Gene records which describe the matched expressions

We obtained about 48,000,000 tokens of training data automatically by using this process using all the 2009 MEDLINE data. This training data includes about 3,000,000 gene mentions.

- ... in the following order: *tna*, *gltC*, *gltS*, *pyrE*; *gltR* is located near ...
- The three genes concerned (designated *entA*, *entB* and *entC*) ...
- Within the hypoglossal nucleus large amounts of *acetylcholinesterase* (*AChE*) activity are ...

Figure 4: False negative examples.

## 2.3 Training Data Expansion

In the previous section, we were able to obtain training data with high-precision by exploiting reference information in the Entrez Gene. However, the resulting data include many false negatives (low-recall), meaning that correct gene names in the data are unannotated. Figure 4 presents an example of missing annotation. In this figure, all gene mentions are shown in italic typeface. The underlined entities were annotated by using the method in Section 2.2, because they were in the Entrez Gene dictionary and this MEDLINE abstract was referred by these entities. However, the entities in italic typeface with no underline were not annotated, because these gene names in Entrez Gene have no link to this MEDLINE abstract. Those expressions became false negatives and became noise for learning. This low-recall problem occurred because no guarantee exists of exhaustiveness in Entrez Gene reference information.

To improve the low-recall while maintaining high-precision, we focused on coordination structures. We assumed that coordinated noun phrases belong to the same semantic class. Figure 5 portrays the algorithm for the annotation expansion based on coordination analysis. We expanded training data annotation using this coordination analysis algorithm to improve annotation recall. This algorithm analyzes whether the words are reachable or not through coordinate tokens such as “,”, “.”, or “and” from initially annotated entities. If the words are reachable and their entities are in the Entrez Gene records (ignoring reference information), then they are annotated.

```
Input: Sequence of sentence tokens $S$, Set of symbols and conjunctions $C$, Dictionary without reference $D$, Set of annotated tokens $A$
Output: Set of Annotated tokens $A$

begin
for $i = 1$ to $|S|$ do
  if $S[i] \in A$ then
    $j \leftarrow i - 2$
    while $1 \leq j \leq |S| \wedge S[j] \in D \wedge S[j] \notin A \wedge S[j+1] \in C$ do
      $A \leftarrow A \cap \{S[j]\}$
      $j \leftarrow j - 2$
    end while
    $j \leftarrow i + 2$
    while $1 \leq j \leq |S| \wedge S[j] \in D \wedge S[j] \notin A \wedge S[j-1] \in C$ do
      $A \leftarrow A \cap \{S[j]\}$
      $j \leftarrow j + 2$
    end while
  end if
end for
Output $A$
end
```

Figure 5: Coordination analysis algorithm.

## 2.4 Self-training

The method described in Section 2.3 reduces false negatives based on coordination structures. However, the training data contain numerous false negatives that cannot be solved through coordination analysis. Therefore, we used a self-training algorithm to automatically correct the training data. In general, a self-training algorithm obtains training data with a small amount of annotated data (seed) and a vast amount of unlabeled text, iterating this process (Zadeh Kaljahi, 2010):

1. Construct a classification model from a seed, then apply the model on the unlabeled text.
2. Annotate recognized expressions as NEs.
3. Add the sentences which contain newly annotated expressions to the seed.

In this way, a self-training algorithm obtains a huge amount of training data.

```
Input: Labeled training data $D$, Machine learning algorithm $A$, Iteration times $n$, Threshold $\theta$
Output: Training data $T_n$

begin
$T_0 \leftarrow$ A seed data from $D$
$i \leftarrow 0$
$D \leftarrow D \backslash T_0$
while $i \neq n$ do
  $M_i \leftarrow$ Construct model with $T_i$
  $U \leftarrow$ Sample some amount of data from $D$
  $L \leftarrow$ Annotate $U$ with model $M_i$
  $U_{new} \leftarrow$ Merge $U$ with $L$ if their confidence values are larger than $\theta$
  $T_{i+1} \leftarrow T_i \cup U_{new}$
  $D \leftarrow D \backslash U$
  $i \leftarrow i + 1$
end while
Output $T_n$
end
```

Figure 6: Self-training algorithm.

In contrast, our case is that we have a large amount of training data with numerous false negatives. Therefore, we adapt a self-training algorithm to revise the training data obtained using the method described in Section 2.3. Figure 6 shows the algorithm. We split the data set ($D$) obtained in Section 2.3 into a seed set ($T_0$) and remaining set ($D \backslash T_0$). Then, we iterate the cycle ($0 \leq i \leq n$):

1. Construct a classification model ($M_i$) trained on the training data ($T_i$).
2. Sample some amount of data ($U$) from the remaining set ($D$).
3. Apply the model ($M_i$) on the sampled data ($U$).
4. Annotate entities ($L$) recognized by this model.
5. Merge newly annotated expressions ($L$) with expressions annotated in Section 2.3 ($U$) if their confidence values are larger than a threshold ($\theta$).
6. Add the merged data ($U_{new}$) to the training data ($T_i$).

In this study, we prepared seed data of 683,000 tokens ($T_0$ in Figure 6). In each step, 227,000 tokens were sampled from the remaining set ($U$).

Because the remaining set $U$ has high precision and low recall, we need not revise NEs that were annotated in Section 2.3. It might lower the quality of the training data to merge annotated entities, thus we used confidence values (Huang and Riloff, 2010) to revise annotations. Therefore, we retain the NE annotations of the remaining set $U$ and overwrite a span of a non-NE annotation only if the current model predicts the span as an NE with high confidence. We compute the confidence of the prediction ($f(x)$) which a token $x$ is predicted as label $y$ as,

$$f(x) = s(x, y) - max(\forall_{z \neq y} s(x, z)).$$

Here, $s(x, y)$ denotes the score (the sum of feature weights) computed using the SVM model described in the beginning of Section 2. A confidence score presents the difference of scores between the predicted (the best) label and the second-best label. The confidence value is computed for each token label prediction. If the confidence value is greater than a threshold ($\theta$) and predicted as an NE of length 1 token (label `S` in IOBES notation), then we revise the NE annotation. When a new NE with multiple tokens (label `B`, `I`, or `E` in IOBES notation) is predicted, we revise the NE annotation if the average of confidence values is larger than a threshold ($\theta$). If a prediction suggests a new entity with multiple tokens $x_i$, ..., $x_j$, then we calculate the average of confidence values as

$$f(x_i, ..., x_j) = \frac{1}{j - i + 1} \sum_{k=i}^{j} f(x_k).$$

The feature set presented in the beginning of Section 2 uses information of the tokens themselves. These features might overfit the noisy seed set, even if we use regularization in training. Therefore, when we use the algorithm of Figure 6, we do not generate token (w) features from tokens themselves but only from tokens surrounding the current token. In other words, we hide information from the tokens of an entity, and learn models using information from surrounding words.

| Method | A | P | R | F1 |
|---|---|---|---|---|
| dictionary matching | 92.09 | 39.03 | 42.69 | 40.78 |
| svm | 85.76 | 10.18 | 23.83 | 14.27 |
| + reference | 93.74 | **69.25** | 39.12 | 50.00 |
| + coordination | 93.97 | 66.79 | 47.44 | 55.47 |
| + self-training | **93.98** | 63.72 | **51.18** | **56.77** |

Table 3: Evaluation results.

## 3 Experiment

The training data automatically generated using the proposed method have about 48,000,000 tokens and 3,000,000 gene mentions. However, we used only about 10% of this data because of the computational cost. For evaluation, we chose to use the BioNLP 2011 Shared Task EPI corpus and evaluation measures described in Section 2.1.

### 3.1 Evaluation of Proposed Methods

In the previous section, we proposed three methods for automatic training data acquisition. We first investigate the effect of these methods on the performance of NER. Table 3 presents evaluation results.

The first method "dictionary matching" simply performs exact string matching with the Entrez Gene dictionary on the evaluation corpus. It achieves a 40.78 F1-measure; this F1-measure will be used as the baseline performance. The second method, as described in Section 2.1, "svm" uses training data generated automatically from the Entrez Gene and unlabeled texts without reference information of the Entrez Gene. The third method, "+ reference" exploits the reference information of the Entrez Gene. This method drastically improves the performance. As shown in Table 3, this model achieves the highest precision (69.25%) with comparable recall (39.12%) to the baseline model with a 50.00 F1-measure. The fourth method, "+ coordination", uses coordination analysis results to expand the initial automatic annotation. Compared to the "+ reference" model, the annotation expansion based on coordination analysis greatly improves the recall (+8.32%) with only a slight decrease of the precision (-2.46%). The last method "+ self-training" applies a self-training technique to improve the performance further. This model achieves the highest recall (51.18%) among all models with a reasonable cost in the precision.

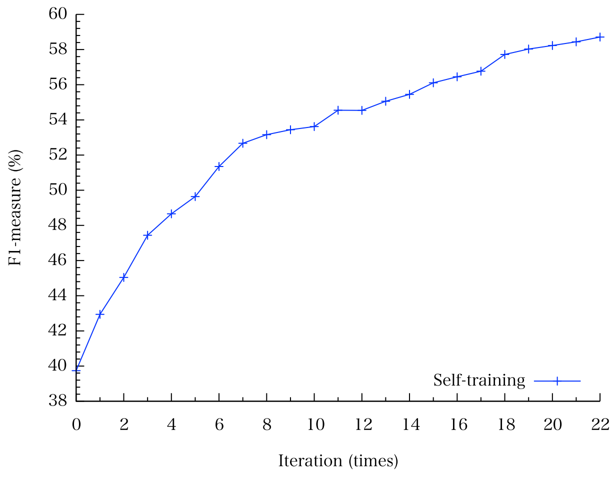


Figure 7: Results of self-training.

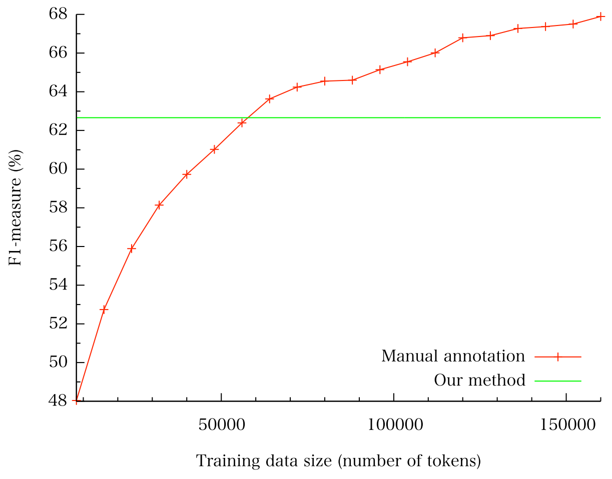


Figure 8: Manual annotation vs. our method.

To analyze the effect of self-training, we evaluated the performance of this model for each iteration. Figure 7 shows the F1-measure of the model as iterations increase. The performance improved gradually. It did not converge even for the last iteration. The size of the training data at the 17th iteration was used in Table 3 experiment. It is the same to the size of the training data for other methods.

## 3.2 Comparison with a Manually Annotated Corpus

NER systems achieving state-of-the-art performance are based mostly on supervised machine learning trained on manually annotated corpus. In this section, we present a comparison of our best-performing NER model with a NER model trained on manually annotated corpus. In addition to the performance comparison, we investigate how much manually annotated data is necessary to outperform our best-performing system. In this experiment, we used only the development data for evaluation because the training data are used for training the NER model.

We split the training data of EPI corpus randomly into 20 pieces and evaluated the performance of the conventional NER system as the size of manually annotated corpus increases. Figure 8 presents the evaluation results. The performance of our our best-performing NER is a 62.66 F1-measure; this is shown as horizontal line in Figure 8. The NER model trained on the all training data of EPI corpus achieves a 67.89 F1-measure. The result shows that our best-performing models achieve comparable performance to that of the NER model when using about 40% (60,000 tokens, 2,000 sentences) of the manually annotated corpus.

## 3.3 Discussion

Although the proposed methods help us to obtain training data automatically with reasonably high quality, we found some shortcomings in these methods. For example, the annotation expansion method based on coordination analysis might find new entities in the training data precisely. However, it was insufficient in the following case.

> *tna* loci, in the following order: *tna*, *gltC*, *gltS*, *pyrE*; *gltR* is located near ...

In this example, all gene mentions are shown in italic typeface. The words with underline were initial annotation with reference information. The surrounding words represented in italic typeface are annotated by annotation expansion with coordination analysis. Here, the first word "tna" shown in italic typeface in this example is not annotated, although its second mention is annotated at the annotation expansion step. We might apply the one sense per discourse (Gale et al., 1992) heuristic to label this case.

Second, the improvement of self-training techniques elicited less than a 1.0 F1-measure. To ascertain the reason for this small improvement, we analyzed the distribution of entity length both origi-

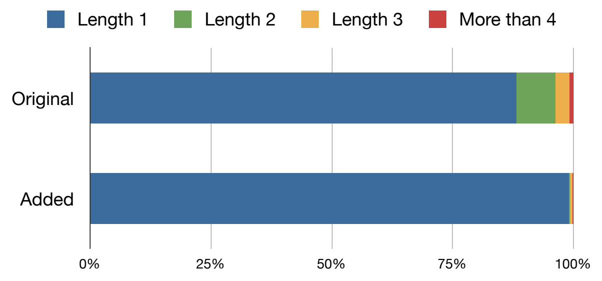


Figure 9: Distribution of entity length.

nally included entities and newly added entities during self-training, as shown in Figure 9. They represent the ratio of entity length to the number of total entities. Figure 9 shows the added distribution of entity length (Added) differs from the original one (Original). Results of this analysis show that self-training mainly annotates entities of the length one and barely recognizes entities of the length two or more. It might be necessary to devise a means to follow the corpus statistics of the ratio among the number of entities of different length as the self-training iteration proceeds.

## 4 Related Work

Our study focuses mainly on achieving high performance NER without manual annotation. Several previous studies aimed at reducing the cost of manual annotations.

Vlachos and Gasperin (2006) obtained noisy training data from FlyBase[1] with few manually annotated abstracts from FlyBase. This study suggested the possibility of acquiring high-quality training data from noisy training data. It used a bootstrapping method and a highly context-based classifiers to increase the number of NE mentions in the training data. Even though the method achieved a high-performance NER in the biomedical domain, it requires curated seed data.

Whitelaw et al. (2008) attempted to create extremely huge training data from the Web using a seed set of entities and relations. In generating training data automatically, this study used context-based tagging. They reported that quite a few good resources (e.g., Wikipedia[2]) listed entities for obtaining training data automatically.

[1]`http://flybase.org/`

[2]`http://www.wikipedia.org/`

Muramoto et al. (2010) attempted to create training data from Wikipedia as a lexical database and blogs as unlabeled text. It collected about one million entities from these sources, but they did not report the performance of the NER in their paper.

## 5 Conclusions

This paper described an approach to the acquisition of huge amounts of training data for high-performance Bio NER automatically from a lexical database and unlabeled text. The results demonstrated that the proposed method outperformed dictionary-based NER. Utilization of reference information greatly improved its precision. Using coordination analysis to expand annotation increased recall with slightly decreased precision. Moreover, self-training techniques raised recall. All strategies presented in the paper contributed greatly to the NER performance.

We showed that the self-training algorithm skewed the length distribution of NEs. We plan to improve the criteria for adding NEs during self-training. Although we obtained a huge amount of training data by using the proposed method, we could not utilize all of acquired training data because they did not fit into the main memory. A future direction for avoiding this limitation is to employ an online learning algorithm (Tong and Koller, 2002; Langford et al., 2009), where updates of feature weights are done for each training instance. The necessity of coordination handling and self-training originates from the insufficiency of reference information in the lexical database, which was not designed to be comprehensive. Therefore, establishing missing reference information from a lexical database to unlabeled texts may provide another solution for improving the recall of the training data.

## References

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237.

Ruihong Huang and Ellen Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 275–285.

Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, pages 1–7.

Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. 2002. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain - Volume 3*, pages 1–8.

John Langford, Lihong Li, and Tong Zhang. 2009. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801.

Hideki Muramoto, Nobuhiro Kaji, Naoki Suenaga, and Masaru Kitsuregawa. 2010. Learning semantic category tagger from unlabeled data. In *The Fifth NLP Symposium for Yung Researchers*. (in Japanese).

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

National Library of Medicine. 2005. Entrez Gene. available at http://www.ncbi.nlm.nih.gov/gene.

National Library of Medicine. 2009. MEDLINE. available at http://www.ncbi.nlm.nih.gov/.

Tomoko Ohta, Jin-Dong Kim, Sampo Pyysalo, Yue Wang, and Jun'ichi Tsujii. 2009. Incorporating genetag-style annotation to genia corpus. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 106–107.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107.

SIGBioMed. 2011. BioNLP 2011 Shared Task. http://sites.google.com/site/bionlpst/.

Lorraine K. Tanabe and W. John Wilbur. 2002. Tagging gene and protein names in biomedical text. *Bioinformatics/computer Applications in The Biosciences*, 18:1124–1132.

Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun 'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics*, volume 3746, pages 382–392.

Andreas Vlachos and Caroline Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 138–145.

Jason Weston and Chris Watkins. 1999. Support vector machines for multi-class pattern recognition. In *ESANN'99*, pages 219–224.

Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. 2008. Web-scale named entity recognition. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 123–132.

Alexander Yeh, Alexander Morgan, Marc Colosimo, and Lynette Hirschman. 2005. Biocreative task 1a: gene mention finding evaluation. *BMC Bioinformatics*, 6(1):S2.

Rasoul Samad Zadeh Kaljahi. 2010. Adapting self-training for semantic role labeling. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 91–96.